

LAPORAN UJIAN AKHIR SEMESTER
“IMPLEMENTASI KONSEP OOP DALAM PEMBUATAN PROGRAM KLINIK
SEDERHANA MENGGUNAKAN JAVA”
UNTUK MEMENUHI TUGAS MATA KULIAH PEMROGRAMAN BERBASIS OBJEK



Disusun Oleh :
Kelompok 2

- | | |
|-------------------------------|---------------|
| 1. Azzelya Rosya Denovya | 2310631250041 |
| 2. Mochamad Ridwan Al Anshori | 2310631250062 |
| 3. Tito Adi Prasetyo | 2310631250035 |
| 4. Nazwa Chandra Febriyanti | 2310631250025 |

Kelas: 4B Sistem Informasi
Dosen Pengampu: Taufik Ridwan S.T, M.T.

PROGAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG 2024/2025

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan Ujian Akhir Semester dengan judul “Implementasi Konsep OOP dalam Pembuatan Program Klinik Sederhana Menggunakan Java” tepat waktu.

Laporan ini disusun untuk memenuhi tugas Mata Kuliah Pemrograman Berorientasi Objek, dengan fokus pada pengembangan aplikasi klinik yang mencakup fitur seperti pendaftaran pasien, manajemen dokter, jadwal praktik, resep obat, dan pengelolaan ruangan serta obat. Seluruh fitur dikembangkan menggunakan konsep Object Oriented Programming dalam bahasa Java.

Kami mengucapkan terima kasih kepada Bapak Taufik Ridwan, S.T., M.T. selaku dosen pengampu, serta semua pihak yang telah memberikan dukungan. Semoga laporan ini bermanfaat dan dapat menjadi referensi bagi pengembangan aplikasi serupa di masa depan.

Karawang, 26 Mei 2025

Kelompok 2

Daftar Isi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin berkembang pesat, pemanfaatan program berbasis GUI (Graphical User Interface) menjadi kebutuhan penting di berbagai sektor, termasuk dalam bidang layanan kesehatan seperti klinik. Klinik modern membutuhkan sistem yang efisien, mudah digunakan, dan berbasis GUI untuk mengelola berbagai aspek operasional, mulai dari pendaftaran pasien, manajemen dokter, jadwal praktik, hingga pengelolaan obat dan ruangan.

Pengembangan program klinik sederhana dengan pendekatan Object-Oriented Programming (OOP) menggunakan bahasa pemrograman Java menjadi solusi yang tepat. Konsep OOP memungkinkan pengembang untuk memodelkan sistem secara modular, dengan memisahkan data dan fungsionalitas dalam bentuk kelas-kelas, sehingga memudahkan pengelolaan dan pengembangan aplikasi dalam jangka panjang. Java sebagai bahasa pemrograman yang stabil dan multiplatform juga sangat mendukung pembuatan program GUI yang handal dan mudah dioperasikan.

Program klinik sederhana yang dikembangkan dalam proyek ini dirancang untuk mencakup beberapa fitur utama, seperti pendaftaran pasien, pengelolaan data dokter, penjadwalan praktik, pembuatan resep obat, serta pengaturan ruangan dan stok obat. Setiap komponen tersebut diimplementasikan dalam bentuk objek, yang saling berinteraksi sesuai prinsip OOP. Selain itu, integrasi fitur CRUD (Create, Read, Update, Delete) juga diterapkan untuk memudahkan admin dan pengguna dalam mengelola data klinik secara menyeluruh.

Melalui laporan ini, akan dibahas secara detail bagaimana konsep OOP diimplementasikan dalam pengembangan program klinik sederhana menggunakan Java. Diharapkan, program ini dapat menjadi solusi efektif dalam membantu proses operasional di klinik serta menjadi media pembelajaran yang aplikatif dalam memahami pemrograman berorientasi objek.

1.2 Rumusan Masalah

Dalam rumusan masalah ini, akan dibahas secara mendalam beberapa aspek yang berkaitan dengan pengembangan program klinik sederhana. Adapun aspek-aspek yang akan dibahas meliputi:

- Bagaimana implementasi konsep Object-Oriented Programming (OOP) dalam pengembangan program klinik sederhana menggunakan bahasa pemrograman Java?
- Bagaimana efektivitas dan efisiensi penggunaan program klinik sederhana dalam membantu proses administrasi dan pelayanan klinik?
- Bagaimana proses pendaftaran pasien dilakukan oleh admin, dan bagaimana data pasien tercatat dan tersimpan dalam sistem?
- Bagaimana sistem mendaftarkan dan menampilkan jadwal praktik, jumlah pasien, jumlah dokter, dan jumlah ruangan.

Dengan rumusan masalah ini, diharapkan dapat memberikan gambaran yang jelas mengenai penerapan konsep OOP dalam pengembangan program klinik, serta bagaimana program ini dapat meningkatkan efisiensi dan efektivitas dalam operasional pelayanan kesehatan di lingkungan klinik.

1.3 Tujuan Masalah

Dalam laporan ini, akan diuraikan secara rinci tujuan-tujuan yang ingin dicapai dari implementasi program klinik sederhana. Tujuan tersebut antara lain:

- Menerapkan konsep Object-Oriented Programming (OOP) dalam pengembangan program klinik sederhana menggunakan bahasa pemrograman Java.
- Meningkatkan efektivitas dan efisiensi pengelolaan layanan dan administrasi di klinik melalui sistem informasi yang terintegrasi.
- Mempermudah admin dalam proses pendaftaran pasien dan pengelolaan data, seperti menambahkan, melihat, mengubah, dan menghapus informasi pasien.
- Menyediakan fitur pengaturan dan tampilan jadwal praktik dokter, jumlah pasien, jumlah dokter, dan jumlah ruangan, serta pengelolaan informasi ruangan klinik secara sistematis.

Dengan tujuan-tujuan ini, diharapkan program klinik sederhana dapat menjadi solusi yang efektif dalam pengelolaan operasional klinik dan mendukung peningkatan kualitas pelayanan kesehatan.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Object-Oriented Programming (OOP)

Object-Oriented Programming (OOP) adalah paradigma pemrograman yang berfokus pada pembuatan objek yang mengandung data (atribut) dan perilaku (metode). Konsep utama OOP meliputi:

1. Kelas (Class): Blueprint untuk membuat objek. Contoh: Pasien, Dokter, Obat.
2. Objek (Object): Instansi dari kelas yang memiliki data dan metode. Contoh: pasien1 dengan nama "Azzelya".
3. Enkapsulasi (Encapsulation): Membungkus data dan metode dalam kelas, serta mengontrol akses melalui modifier (public, private, protected).
4. Inheritance (Pewarisan): Kelas anak mewarisi sifat kelas parent. Contoh: Dokter bisa mewarisi Karyawan.
5. Polimorfisme (Polymorphism): Kemampuan objek untuk memiliki banyak bentuk. Contoh: metode tampilkanInfo() di kelas Pasien dan Dokter dengan implementasi berbeda.
6. Abstraksi (Abstraction): Menyembunyikan detail kompleks dan menampilkan fungsionalitas esensial.

Java dipilih karena mendukung penuh OOP, multiplatform, dan memiliki library GUI seperti Swing/JavaFX untuk antarmuka pengguna.

2.2 Fungsi Utama Perangkat Lunak

Berikut adalah penjelasan mengenai fitur utama dari perangkat lunak yang dibuat untuk Program Klinik Sederhana:

1. Pendaftaran Pasien

Fitur ini memungkinkan admin untuk mendaftarkan pasien baru ke sistem. Data yang dimasukkan bisa berupa nama, umur, alamat, nomor kontak, dan keluhan awal.

2. List Obat

Menampilkan daftar semua obat yang tersedia di klinik, termasuk nama obat, jenis, dan stok.

3. List Dokter

Menampilkan informasi dokter yang bekerja di klinik, seperti nama, spesialisasi, dan jadwal praktik.

4. Jadwal Praktik

Menyimpan dan menampilkan jadwal praktik dokter. Dapat digunakan oleh admin untuk pengaturan, dan oleh dokter untuk melihat jadwal dirinya.

5. Resep Obat

Dokter dapat membuat resep obat berdasarkan data pasien. Resep berisi nama pasien, obat yang diresepkan, dan dosisnya.

6. List Ruangan

Menampilkan daftar ruangan yang ada di klinik, seperti ruang pemeriksaan, ruang tunggu, ruang perawatan, dll.

Dengan adanya fitur-fitur tersebut, penggunaan program klinik sederhana dapat membantu meningkatkan efektivitas dan efisiensi pelayanan kesehatan. Admin dapat dengan mudah mendaftarkan pasien baru dan mengelola informasi penting seperti data dokter, jadwal praktik, serta ketersediaan obat dan ruangan. Dokter pun terbantu dengan fitur resep obat yang memungkinkan penulisan resep secara sistematis dan terdokumentasi. Selain itu, pasien dapat dilayani dengan lebih cepat karena semua informasi tersedia secara terstruktur dan terpusat. Secara keseluruhan, aplikasi ini mampu mengurangi kesalahan pencatatan manual dan mendukung kelancaran operasional klinik sehari-hari.

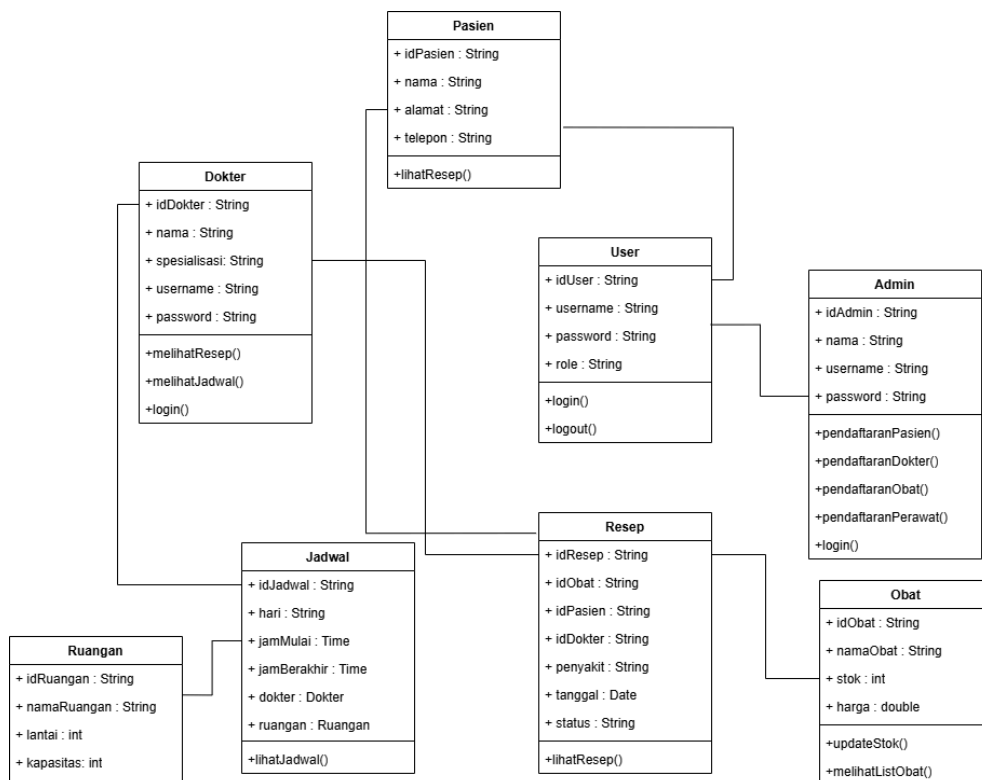
BAB III

PEMBAHASAN

3.1 Perancangan Sistem (UML)

Dalam konteks perancangan program klinik sederhana, diagram pada Unified Modeling Language (UML) digunakan untuk memvisualisasikan struktur dan perilaku sistem secara menyeluruh. Melalui berbagai jenis diagram seperti class diagram, use case, activity diagram, sequence diagram, dan interaksi aktor dengan sistem, perancangan menjadi lebih terarah dan terdokumentasi dengan baik.

3.1.1 Class Diagram



Class diagram ini menggambarkan struktur utama dari sistem informasi klinik sederhana, yang terdiri dari beberapa kelas beserta atribut dan metode yang dimilikinya. Setiap kelas mewakili entitas yang memiliki peran penting dalam sistem.

- **Pasien** menyimpan data pasien seperti id, nama, alamat, dan nomor telepon. Pasien juga memiliki fungsi untuk melihat resep.
- **Dokter** memiliki atribut seperti id, nama, spesialisasi, username, dan password. Dokter dapat melakukan login serta melihat resep dan jadwal praktiknya.

- Class diagram ini menunjukkan bagaimana masing-masing entitas saling berhubungan dan berinteraksi dalam sistem. Hubungan antar kelas juga mendukung proses bisnis utama klinik, seperti pendaftaran pasien, pemberian resep, pengaturan jadwal dokter, hingga pengelolaan data obat dan ruangan.

```

    usecaseDiagram
        actor Admin
        actor Dokter
        usecase Login
        usecase Kelola_data as Kelola data
        usecase Pendaftaran_Pasien as Pendaftaran Pasien
        usecase Pendaftaran_Dokter as Pendaftaran Dokter
        usecase Pendaftaran_Perawat as Pendaftaran Perawat
        usecase Pendaftaran_Obat as Pendaftaran Obat
        usecase Lihat_data as Lihat data
        usecase Melihat_List_Pasien as Melihat List Pasien
        usecase Melihat_List_Dokter as Melihat List Dokter
        usecase Melihat_List_Perawat as Melihat List Perawat
        usecase Melihat_Resep_Pasien as Melihat Resep Pasien
        usecase Melihat_Jadwal_Praktik as Melihat Jadwal Praktik
        usecase Membuat_Resep_Pasien as Membuat resep pasien
        usecase Kelola_Resep_Pasien as Kelola resep pasien
        usecase Melihat_Jadwal_Praktik_Dokter as Melihat jadwal praktik

        Admin --> Login
        Admin --> Kelola_data
        Admin --> Lihat_data
        Dokter --> Login
        Dokter --> Melihat_Jadwal_Praktik_Dokter

        Kelola_data -|> Pendaftaran_Pasien
        Kelola_data -|> Pendaftaran_Dokter
        Kelola_data -|> Pendaftaran_Perawat
        Kelola_data -|> Pendaftaran_Obat

        Lihat_data -|> Melihat_List_Pasien
        Lihat_data -|> Melihat_List_Dokter
        Lihat_data -|> Melihat_List_Perawat
        Lihat_data -|> Melihat_Resep_Pasien
        Lihat_data -|> Melihat_Jadwal_Praktik

        Kelola_Resep_Pasien -|> Membuat_Resep_Pasien
        Kelola_Resep_Pasien -|> Melihat_Jadwal_Praktik_Dokter

        Kelola_data ..> Pendaftaran_Pasien : <<include>>
        Kelola_data ..> Pendaftaran_Dokter : <<include>>
        Kelola_data ..> Pendaftaran_Perawat : <<include>>
        Kelola_data ..> Pendaftaran_Obat : <<include>>

        Lihat_data ..> Melihat_List_Pasien : <<include>>
        Lihat_data ..> Melihat_List_Dokter : <<include>>
        Lihat_data ..> Melihat_List_Perawat : <<include>>
        Lihat_data ..> Melihat_Resep_Pasien : <<include>>
        Lihat_data ..> Melihat_Jadwal_Praktik : <<include>>

        Kelola_Resep_Pasien ..> Membuat_Resep_Pasien : <<include>>
        Kelola_Resep_Pasien ..> Melihat_Jadwal_Praktik_Dokter : <<include>>
  
```

Use case diagram ini menggambarkan interaksi antara aktor (pengguna) dan sistem klinik, yang mencakup dua aktor utama, yaitu Admin dan Dokter. Diagram ini menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh masing-masing aktor.

Aktor:

- Admin: Memiliki akses penuh untuk mengelola dan melihat data di sistem.
- Dokter: Berperan dalam melihat jadwal praktik dan mengelola resep pasien.

Fitur yang Diakses Admin:

- Login: Admin harus masuk ke sistem terlebih dahulu.
- Kelola Data: Admin dapat mengakses menu kelola data, yang mencakup: Pendaftaran Pasien, Pendaftaran Dokter, Pendaftaran Perawat, Pendaftaran Obat.
- Lihat Data: Admin juga dapat melihat berbagai daftar informasi seperti: List Pasien, List Dokter, List Perawat, Jadwal Praktik, dan Resep Pasien.

Relasi <<include>> menunjukkan bahwa fungsi-fungsi seperti pendaftaran dan melihat daftar merupakan bagian dari proses utama *Kelola Data* dan *Lihat Data*.

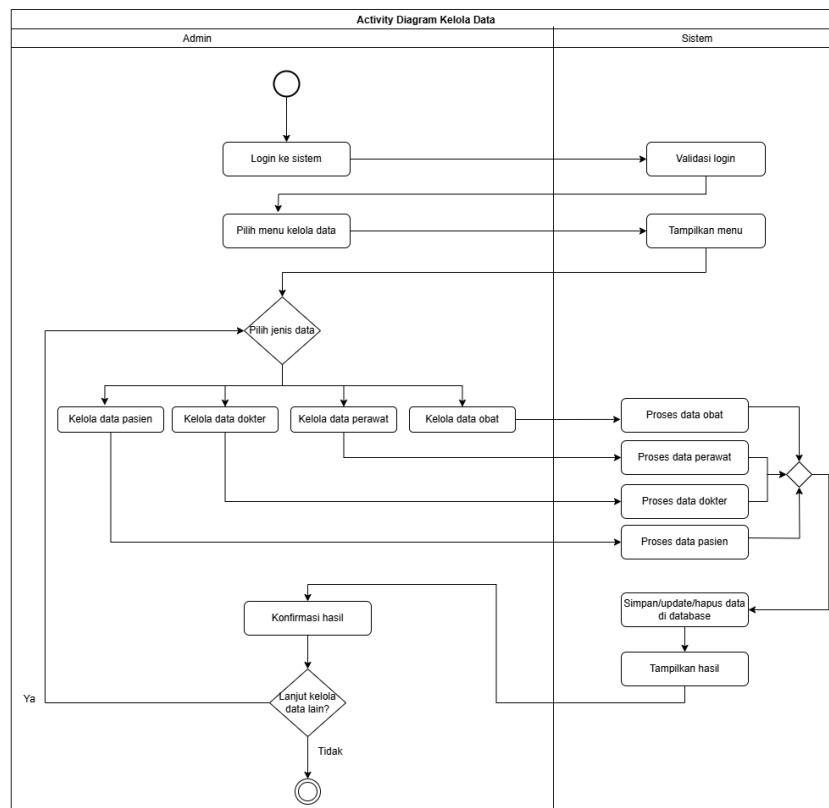
Fitur yang Diakses Dokter:

- Login: Dokter juga harus login terlebih dahulu.
- Melihat Jadwal Praktik: Dokter dapat melihat jadwal tugas/praktik mereka.
- Kelola Resep Pasien: Dokter dapat membuat resep dan melihat resep pasien yang pernah dibuat.

Dalam *Kelola Resep Pasien*, terdapat dua aktivitas yang termasuk di dalamnya, yaitu *Membuat Resep Pasien* dan *Melihat Resep Pasien*.

3.1.3 Activity Diagram

a. Activity Diagram Kelola Data

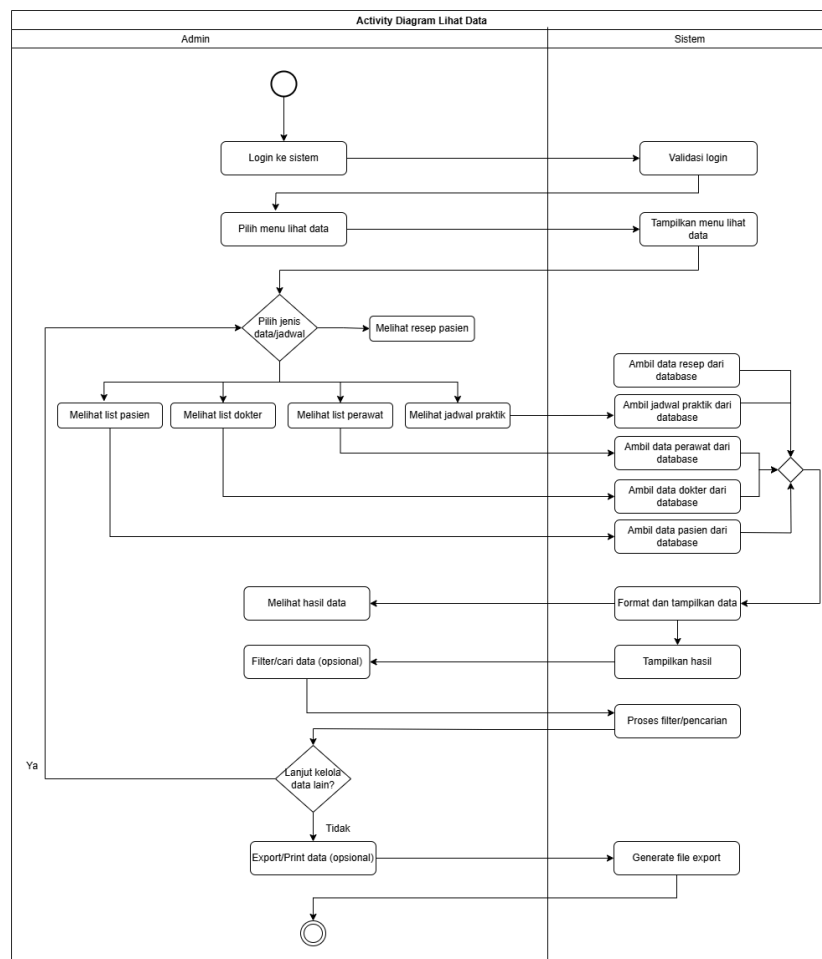


Proses dimulai saat Admin login ke sistem, kemudian sistem melakukan validasi login. Setelah berhasil login, Admin memilih menu kelola data, lalu sistem menampilkan menu yang tersedia. Admin dapat memilih salah satu dari beberapa jenis data untuk dikelola:

- Kelola data pasien → sistem memproses data pasien sesuai aksi (tambah, ubah, hapus).
- Kelola data dokter → sistem memproses data dokter sesuai aksi.
- Kelola data perawat → sistem memproses data perawat sesuai aksi.
- Kelola data obat → sistem memproses data obat sesuai aksi.

Setelah data diproses, sistem akan menyimpan/update/hapus data di database dan menampilkan hasilnya. Admin diminta untuk melakukan konfirmasi hasil, lalu memutuskan apakah ingin melanjutkan ke pengelolaan data lain atau mengakhiri proses. Semua aktivitas yang berhasil akan diakhiri dengan notifikasi hasil dari sistem.

b. Activity Diagram Lihat Data



Proses dimulai saat Admin login ke sistem, kemudian sistem melakukan validasi login. Setelah berhasil login, Admin memilih menu lihat data, lalu sistem menampilkan pilihan menu lihat data. Admin kemudian diminta untuk memilih jenis data atau jadwal yang ingin dilihat. Ada beberapa pilihan jenis data/jadwal yang tersedia:

- Melihat resep pasien → sistem mengambil data resep dari database.
- Melihat list pasien → sistem mengambil data pasien dari database.
- Melihat list dokter → sistem mengambil data dokter dari database.
- Melihat list perawat → sistem mengambil data perawat dari database.
- Melihat jadwal praktik → sistem mengambil data jadwal praktik dari database.

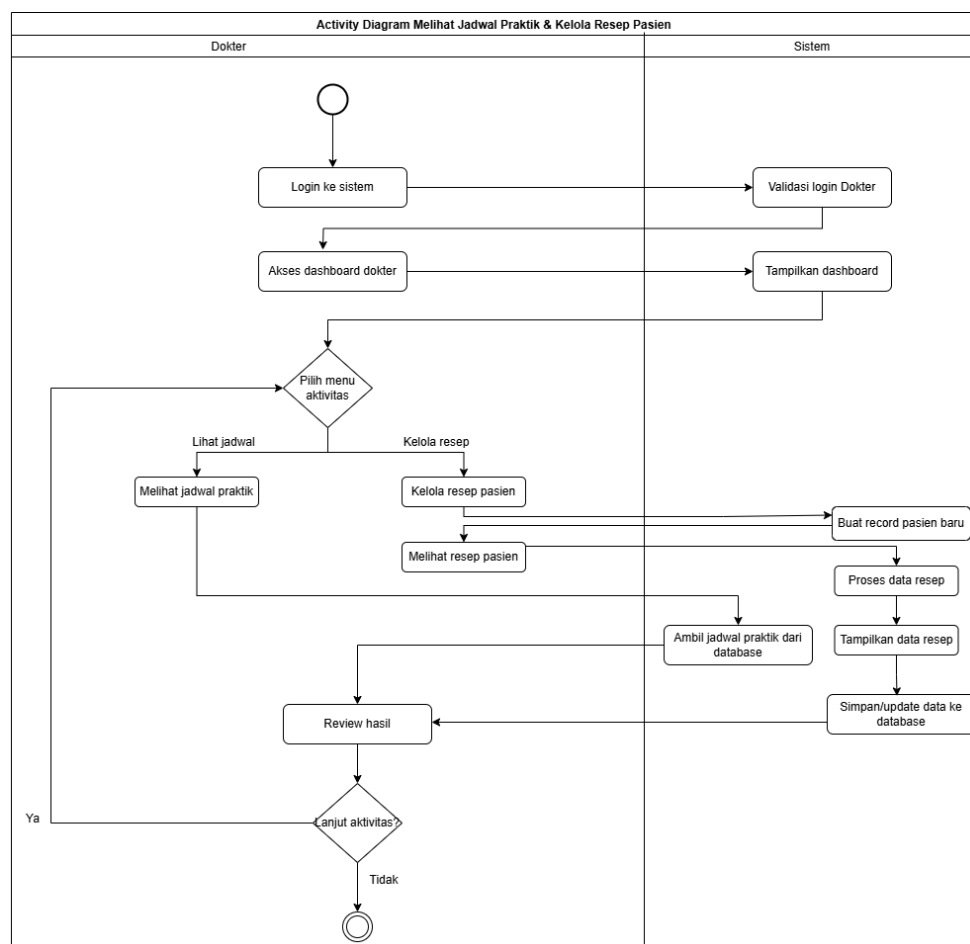
Setelah data yang diminta diambil, sistem akan melakukan format dan menampilkan data tersebut kepada Admin.

Admin kemudian dapat:

- Melihat hasil data, dan
- (Opsional) Melakukan filter atau pencarian data tertentu, yang akan diproses dan ditampilkan oleh sistem.

Setelah itu, Admin memutuskan apakah ingin melanjutkan untuk mengelola data lain. Jika ya, maka proses akan kembali ke pemilihan jenis data/jadwal. Jika tidak, maka Admin dapat mengekspor atau mencetak data (opsional), dan proses pun berakhir.

c. Activity Diagram Melihat Jadwal Praktik & Kelola Resep Pasien



Proses dimulai saat Dokter login ke sistem, kemudian sistem melakukan validasi login Dokter. Setelah berhasil login, Dokter mengakses dashboard, lalu sistem menampilkan tampilan dashboard. Selanjutnya, Dokter diminta untuk memilih menu aktivitas. Terdapat dua pilihan utama:

1. Lihat jadwal praktik

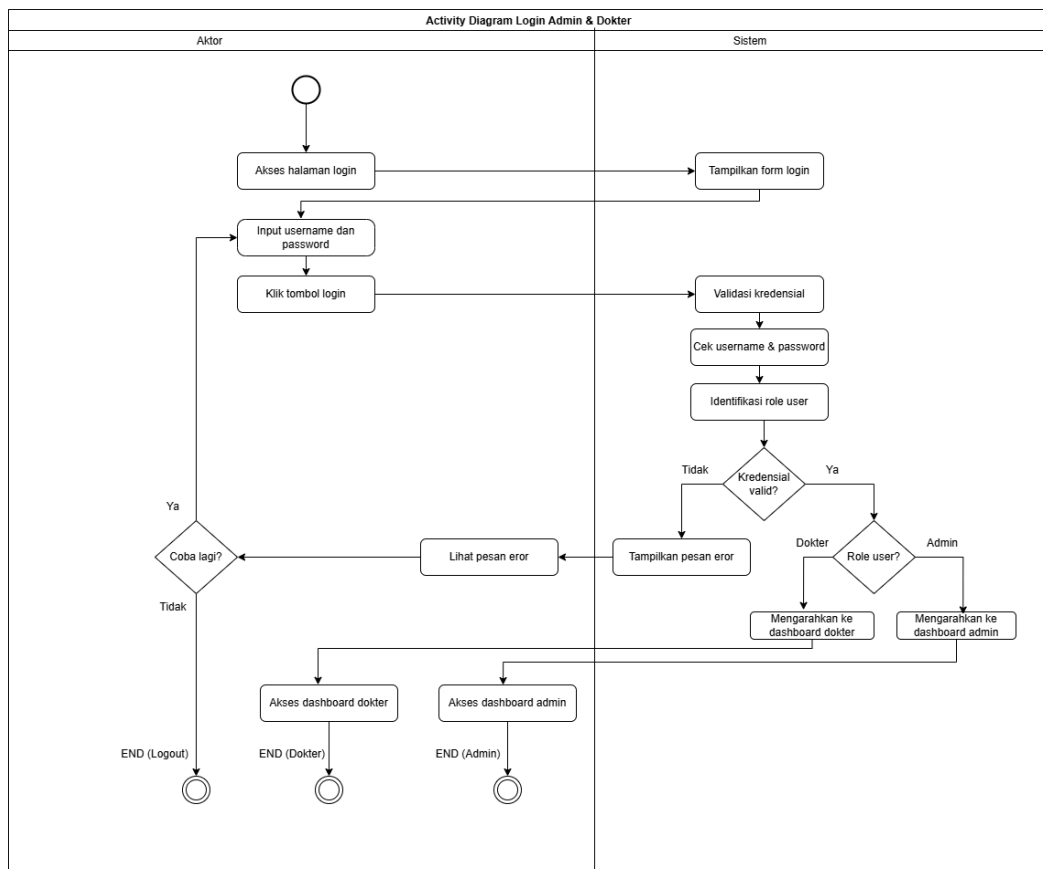
Dokter memilih untuk melihat jadwal praktik, kemudian sistem mengambil data jadwal praktik dari database, dan menampilkannya kepada dokter.

2. Kelola resep pasien

Dokter memilih menu kelola resep, lalu Dokter dapat melihat resep pasien atau mengelola resep pasien. Sistem akan memeriksa apakah perlu membuat record pasien baru (jika belum ada), kemudian memproses data resep yang diberikan dokter, Lalu menampilkan data resep, dan menyimpan atau mengupdate data ke database.

Setelah aktivitas selesai, Dokter akan melakukan review terhadap hasil aktivitas. Kemudian sistem menanyakan apakah dokter ingin melanjutkan aktivitas lain. Jika ya, maka proses kembali ke pemilihan menu aktivitas. Jika tidak, maka proses berakhir.

d. Activity Diagram Login Admin & Dokter



Proses dimulai saat aktor (baik Admin maupun Dokter) mengakses halaman login, lalu sistem menampilkan form login. Pengguna kemudian memasukkan username dan

password, lalu menekan tombol login. Setelah itu, sistem melakukan validasi kredensial dengan:

1. Memeriksa kecocokan username dan password.
2. Mengidentifikasi peran (role) dari user.

Selanjutnya, sistem memeriksa apakah kredensial valid:

- Jika tidak valid, sistem akan menampilkan pesan error, dan pengguna akan diarahkan untuk memutuskan apakah ingin mencoba login lagi:
 - Jika ya, kembali ke proses input login.
 - Jika tidak, maka proses login diakhiri (logout).
- Jika valid, sistem akan mengecek role user:
 - Jika role-nya dokter, maka sistem akan mengarahkan ke dashboard dokter, dan proses login berakhir (END Dokter).
 - Jika role-nya admin, maka sistem akan mengarahkan ke dashboard admin, dan proses login berakhir (END Admin).

3.1.4 Sequence Diagram

a. sequence diagram kelola data

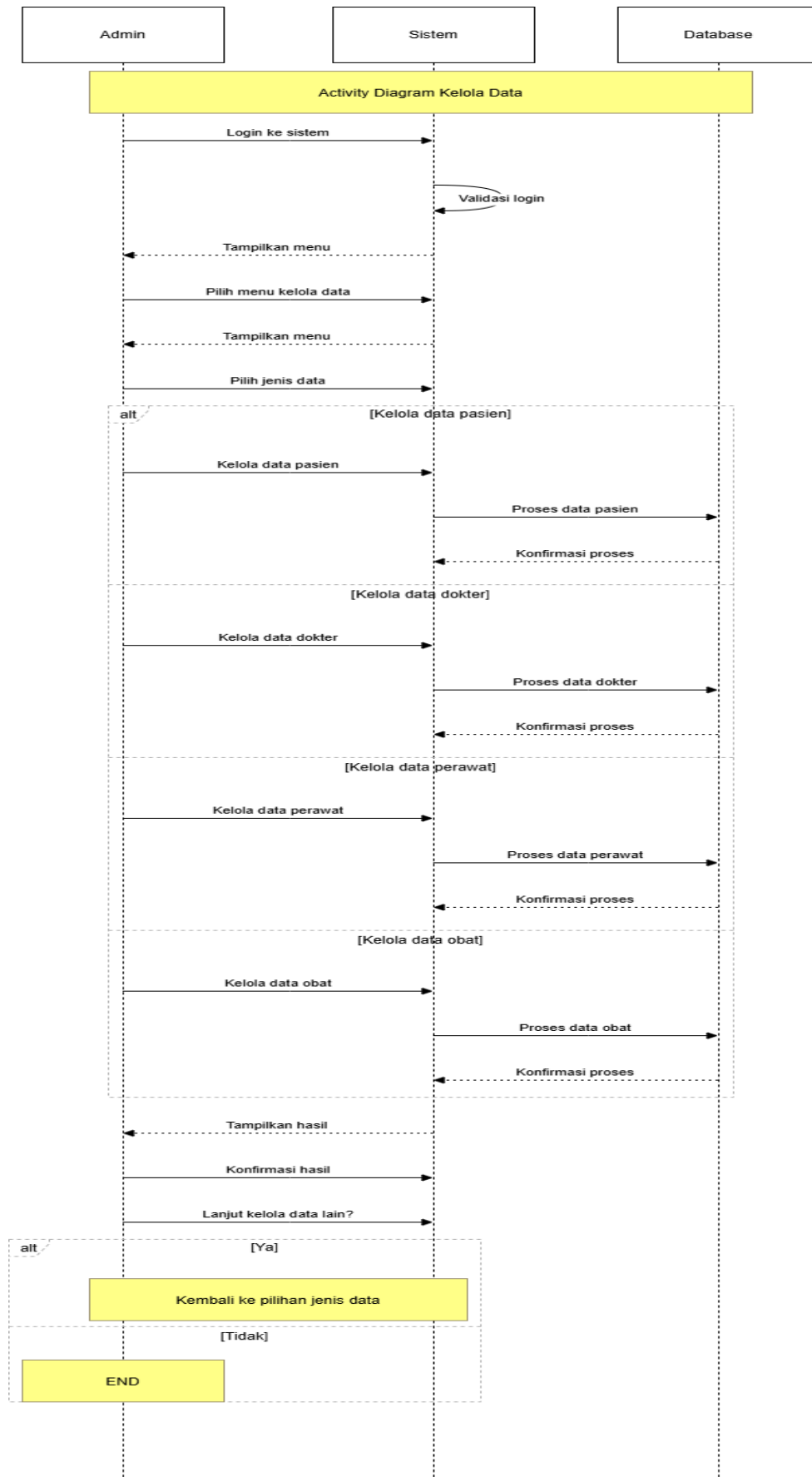


Diagram ini menunjukkan fungsi admin dalam mengelola berbagai jenis data dalam sistem dengan fokus pada empat kategori data utama:

- **Fitur Utama Admin:**

1. Login dan Validasi: Admin login ke sistem dengan validasi yang berhasil
2. Menu Kelola Data: Admin memilih menu kelola data dari dashboard utama

- **Jenis Data yang Dikelola:**

1. Kelola Data Pasien

- Admin memilih opsi "Kelola data pasien"
- Sistem memproses data pasien ke database
- Database memberikan konfirmasi proses berhasil

2. Kelola Data Dokter

- Admin memilih opsi "Kelola data dokter"
- Sistem melakukan pemrosesan informasi dokter
- Database mengkonfirmasi penyimpanan data dokter

3. Kelola Data Perawat

- Admin memilih opsi "Kelola data perawat"
- Sistem mengelola dan memproses data perawat
- Database memberikan konfirmasi proses data perawat

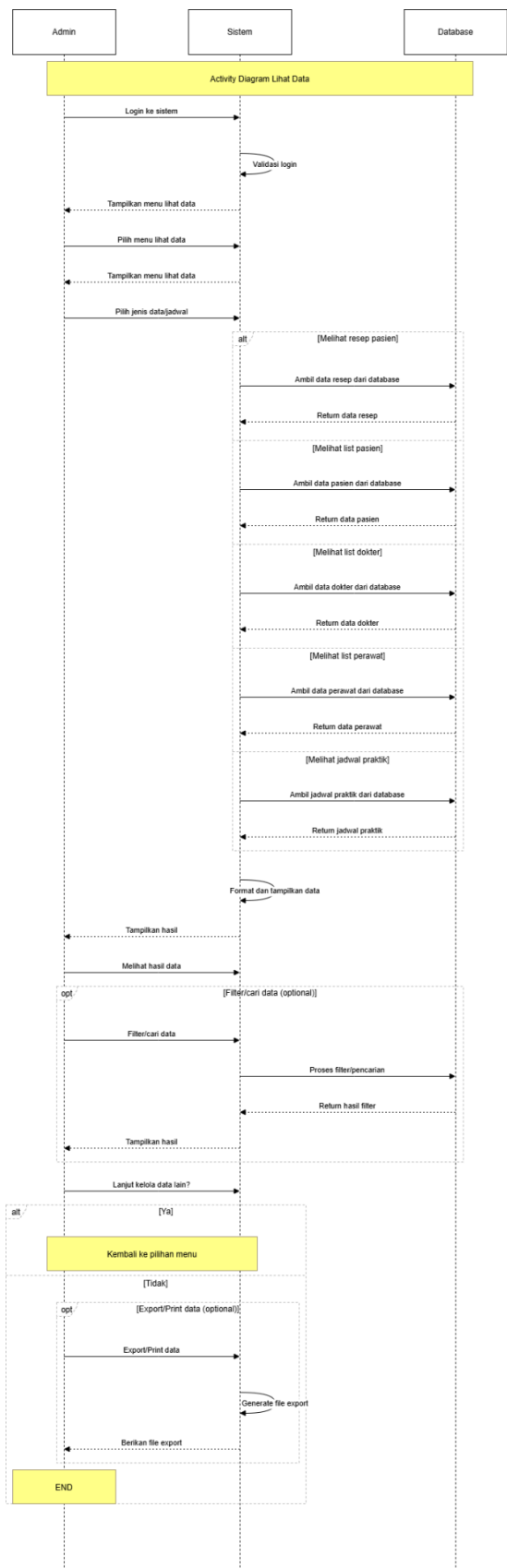
4. Kelola Data Obat

- Admin memilih opsi "Kelola data obat"
- Sistem memproses data obat dan informasi farmasi
- Database mengkonfirmasi penyimpanan data obat berhasil

- **Alur Kerja:**

1. Setelah login, admin mengakses menu kelola data
2. Admin memilih jenis data yang akan dikelola (pasien, dokter, perawat, atau obat)
3. Sistem memproses data sesuai pilihan ke database
4. Database memberikan konfirmasi hasil proses
5. Admin dapat melanjutkan mengelola jenis data lain
6. Proses berakhir ketika admin selesai mengelola semua data yang diperlukan

b. Sequence diagram lihat data



c. Sequence diagram melihat jam praktik

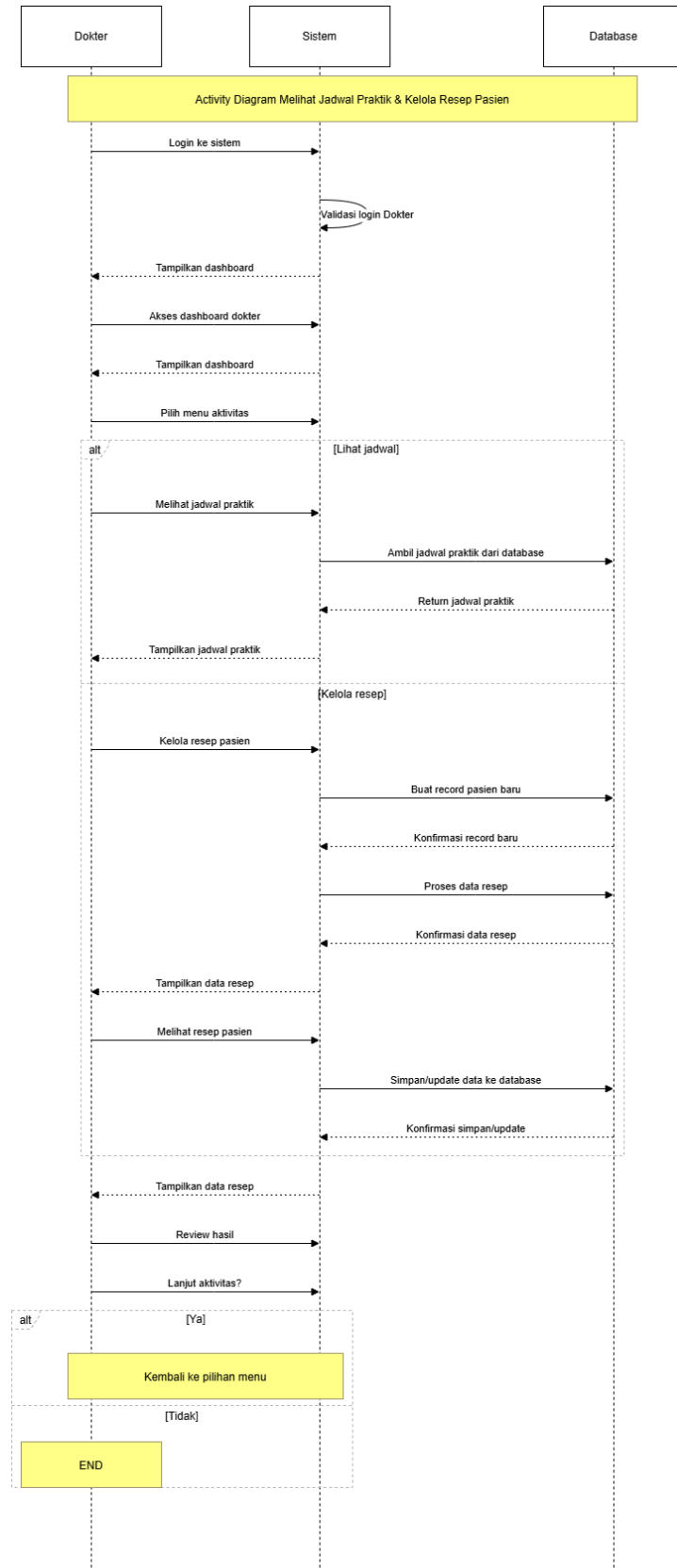


Diagram ini menjelaskan dua fungsi utama dokter dalam sistem:

a. Melihat Jadwal Praktik

1. Login Validasi: Dokter login ke sistem dengan validasi yang berhasil
2. Akses Dashboard: Sistem menampilkan dashboard dokter
3. Pilih Menu Aktivitas: Dokter memilih menu untuk melihat jadwal praktik
4. Ambil Data Jadwal: Sistem mengambil data jadwal praktik dari database
5. Tampilkan Jadwal: Sistem menampilkan jadwal praktik dokter

b. Kelola Resep Pasien

1. Pilih Kelola Resep: Dokter memilih menu kelola resep pasien
2. Buat Record Baru: Sistem memungkinkan pembuatan record resep baru
3. Konfirmasi Data: Sistem mengkonfirmasi data resep baru
4. Proses ke Database: Data resep diproses dan disimpan ke database
5. Update Konfirmasi: Database memberikan konfirmasi penyimpanan data
6. Tampilkan Hasil: Sistem menampilkan data resep yang telah disimpan
7. Review dan Kontinuitas: Dokter dapat melakukan review hasil dan melanjutkan aktivitas lain

d. Sequence diagram login admin

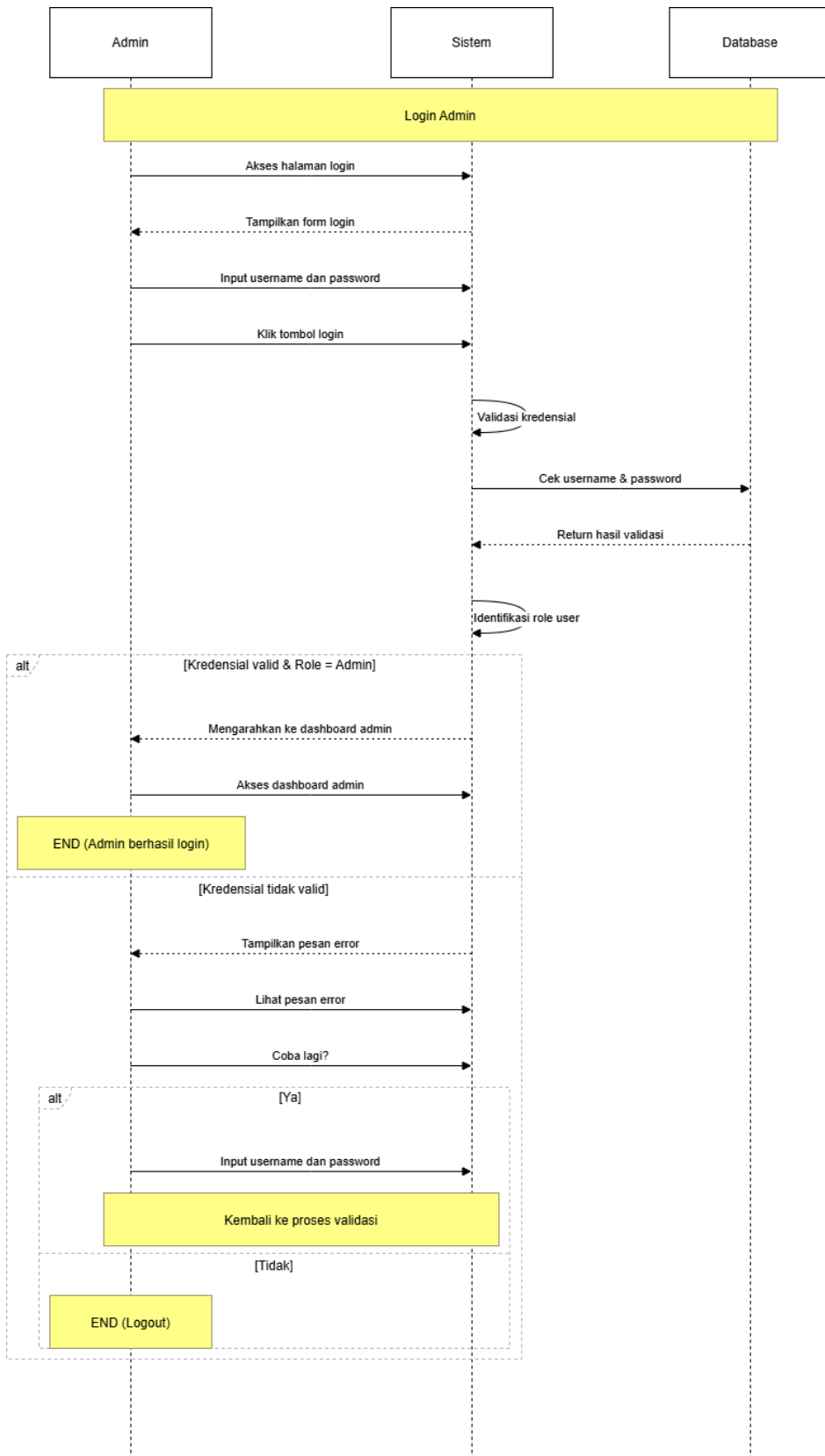


Diagram ini menunjukkan alur proses autentikasi untuk pengguna dengan role admin. Prosesnya serupa dengan login dokter namun dengan akses ke dashboard admin:

1. Akses Halaman Login: Admin mengakses halaman login sistem
2. Input Kredensial: Admin memasukkan username dan password
3. Validasi Database: Sistem memverifikasi kredensial melalui database
4. Identifikasi Role: Sistem mengkonfirmasi role user sebagai admin
5. Redirect Dashboard: Jika berhasil, sistem mengarahkan ke dashboard admin
6. Error Handling: Penanganan kesalahan jika kredensial tidak valid
7. Logout Process: Proses logout dari sistem

e. Sequence diagram login dokter

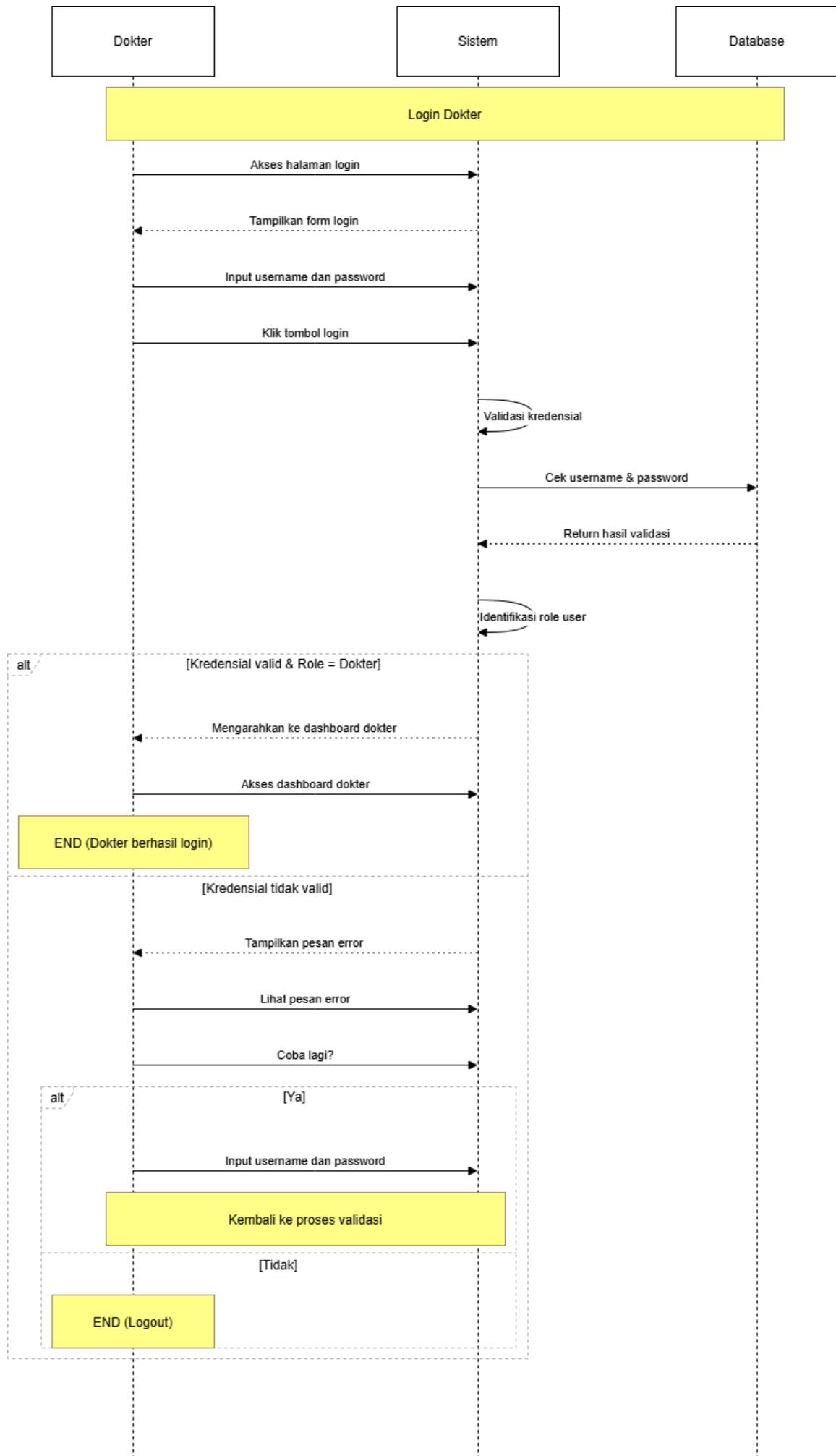


Diagram ini menggambarkan alur proses autentikasi untuk pengguna dengan role dokter. Proses dimulai ketika dokter mengakses halaman login sistem. Berikut adalah tahapan prosesnya:

1. Inisiasi Login: Dokter mengakses halaman login dan sistem menampilkan form login
2. Input Kredensial: Dokter memasukkan username dan password, kemudian mengklik tombol login
3. Validasi Kredensial: Sistem melakukan pengecekan username dan password ke database
4. Pengembalian Hasil: Database mengembalikan hasil validasi ke sistem
5. Identifikasi Role: Sistem mengidentifikasi role user sebagai dokter
6. Akses Dashboard: Jika kredensial valid dan role adalah dokter, sistem mengarahkan ke dashboard dokter
7. Penanganan Error: Jika kredensial tidak valid, sistem menampilkan pesan error dan meminta user untuk mencoba lagi
8. Logout: Proses berakhir dengan kemungkinan logout dari sistem